

Configuration management as a multi-cloud enabler

Bart Vanbrabant and Wouter Joosen
iMinds-DistriNet, Dept. of Computer Science, K.U.Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
{bart.vanbrabant, wouter.joosen}@cs.kuleuven.be

ABSTRACT

The deployment of applications in hybrid and multi-cloud environments avoids vendor lock-in and reliance on a single cloud provider. The virtual compute infrastructures of IaaS provides enables enterprises to build large scale, geographically distributed applications. Each IaaS provider offers virtual machines that can communicate with each other over a virtual network. However, each IaaS provider exposes subtle differences in the support for virtual machines, storage, networking, routing, etc. In this position paper we argue that configuration management is an enabler for multi-cloud environments by making abstraction of the above mentioned differences in the IaaS platforms, through configuration management. We illustrate this claim by using configuration management to address the challenges associated with multi-cloud environments.

Categories and Subject Descriptors

C.2.3 [Computer - Communication Networks]: Network Operations; C.2.4 [Computer - Communication Networks]: Distributed Systems

Keywords

IaaS, multi-cloud, configuration management, distributed systems

1. INTRODUCTION

Cloud computing provides elastic resources to its users. Cloud services are often offered in three service levels: IaaS, PaaS and SaaS. The lowest abstraction level in cloud offerings is virtual compute infrastructures (IaaS). Primarily, it provides virtual machines and network virtualization. Platform and Software as a service offers its users a higher level of abstraction at the price of reduced flexibility. This position paper focusses on the IaaS level of cloud computing. IaaS can be private (on-premise), public or a mix of the two in a hybrid cloud or multi-cloud infrastructure. The multi-cloud

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CrossCloud Brokers '14 December 8, 2014, Bordeaux, France

Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-3233-0 ...\$15.00
<http://dx.doi.org/10.1145/2676662.2676672>.

model offers the capability to shift load between a private cloud with fixed capacity and a public cloud with higher costs, yet high elasticity. The goal of a multi-cloud infrastructure is to ensure that cloud capacity is always available and to avoid the reliance on a single provider.

At the IaaS level multi-cloud is straight-forward in the sense that most cloud providers offer virtual machines that can communicate privately with each other and access the internet. Therefore, virtual machines at multiple IaaS providers can communicate with each other and host a single distributed system. However, each Cloud provider offers slightly different services that can be programmed with their own API's. Abstraction layers, such as JClouds [2], offer a single API for multiple IaaS providers. Table 1 gives an overview of available infrastructure services at well known IaaS providers. The table shows that although the basic offering is similar there are subtle difference between IaaS providers. An operational multi-cloud infrastructure requires more than virtual machines at multiple IaaS providers. Operating, deploying and managing, a distributed application in such an environment encounters the following issues:

- Small differences in operating system between providers such 32 vs 64 bit or different types of virtualization (full or para-virtualization).
- The architecture and implementation of many distributed systems assume that components or nodes in the system can communicate securely with each other. This requires that the virtual machines connect to each other through a private network, isolated from other tenants. Additionally, inter-IaaS communication cannot be routed over the public internet without additional security.
- Many distributed systems use discovery mechanisms that rely on broadcast and multicast. These are not supported by most IaaS providers. Many communication libraries solve this by employing IaaS specific workarounds, for example the *aws-ping* and *rackspace-ping* discovery protocols of JGroups [7]. However, they do not work in a multi-cloud environment.
- Each IaaS provider offers a limited number of types (sometimes named “flavors”) of virtual machines. Each type has a specific performance profile (number of cpu's, amount of memory, storage, type of storage, etc.) with its associated pricing.
- Public IaaS providers give virtual machines a public IP address and a hostname. However, the naming con-

vention is highly provider dependent. This complicates management and troubleshooting because these hostnames reveal nothing about its function.

The aforementioned challenges for multi-cloud deployment at the IaaS level can be addressed by adding services to the infrastructure and configuring virtual machines to use them. In this paper we give an overview of how these challenges can be solved by means of adding infrastructure services and configuration. However, this increases the effort and complexity to deploy and manage the multi-cloud infrastructure. Configuration management tools can reduce the increased management complexity by automating the deployment and management of the distributed system.

Next in this paper we give a short overview of configuration management tools. This overview is followed by a description of how additional configuration can solve multi-cloud challenges and how configuration management tools automate this configuration. Then we give a short overview of related work and conclude this position paper.

2. CONFIGURATION MANAGEMENT

Configuration management tools offer a structured approach to system administration automation. These tools exist in a broad range of automation and abstraction capabilities. At one end of the spectrum are tools that merely provide a framework to distribute, schedule and execute *custom imperative scripts* and generate reports from the execution. On the other end of the spectrum are tools that offer a *desired state configuration model* of the managed infrastructure. The latter effectively provide infrastructure as code [8, 6]. In this paper we focus on the tools that use a declarative configuration model for the desired state of the complete infrastructure.

Figure 1 shows a simplified diagram of a reference architecture for *desired state* tools. A central repository stores an input specification that describes the desired configuration of the managed distributed system. One or more translation agents generate the desired state of each of the resources the tool manages on each IaaS and the virtual machines. A deployment agent retrieves the desired state of each of the resources it manages. The deployment agent then enforces the desired state of each of its managed resources.

In a cloud setting configuration management tools are a prerequisite because of the level of automation they help to achieve. Only with sufficient automation the full elasticity of cloud computing can be leveraged. Sufficient automation entails: 1. starting and stopping virtual machines. 2. ensuring that each virtual machine has its basic install (logging, monitoring, etc.) 3. installing, configuring and starting the software services that should run on the virtual machine. 4. integrate the virtual machine in the existing infrastructure: add it to DNS, logging, monitoring and configure it in the existing distributed system. For example, adding an additional node in a webserver cluster to the loadbalancers.

3. MULTI-CLOUD IAAS

The challenges of a multi-cloud IaaS can be addressed by adding additional services and configuration to the infrastructure. In this section we sketch solutions for these challenges and explain how configuration management can automate the management of the proposed solutions.

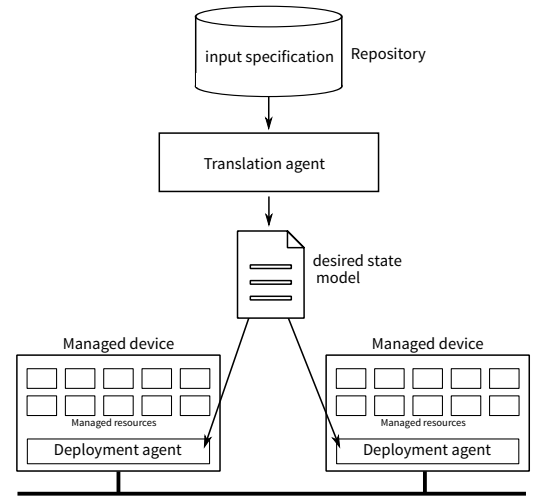


Figure 1: A reference architecture for configuration management tools.

Networking. Most IaaS providers give virtual machines public and private networking. The public network is used to communicate with the internet. The private interface is used to communicate with other services hosted on virtual machines at the same IaaS provider. This network uses non-routable IP addresses, as such the traffic cannot end up on the public Internet. Additionally, most IaaS providers use network virtualization techniques to separate network traffic of their tenants.

In a multi-cloud setting the traffic between virtual machines on different providers is routed over the Internet. A VPN connection can secure Inter-IaaS traffic. It encrypts traffic before it is routed over the Internet. This requires VPN endpoints in each IaaS and routing rules for all virtual machines to route traffic for VM's in other clouds to the VPN endpoint. The limited survey of IaaS providers in table 1 finds that most providers do not offer managed VPN services and some not even custom routing rules.

A solution is to deploy VPN endpoints on a virtual machine in each IaaS. Inside the IaaS we either add routing rules to the gateway of the IaaS provider if this is supported, or deploy routing rules for the private networks of the other IaaS providers on each virtual machine. A configuration model can describe the entire configuration of the VPN endpoints, the required PKI to setup private keys and certificates and routing rules on all virtual machines. Management agents on each virtual machine install the private keys, certificates, VPN software and add routing rules on each virtual machine.

Discovery. Distributed systems (middleware platforms, storage systems, ...) use discovery mechanism to find other nodes of a distributed system. A discovery mechanism reduces configuration complexity because it often only requires a cluster name in the configuration. In many IaaS environments the network virtualization does not support broadcast and multicast. Indeed, many different and IaaS specific discovery mechanisms exist [7, 5]. Multi-cloud environments make this even harder because each IaaS requires a specific discovery mechanism that often uses a storage API and predefined storage keys.

A configuration model that models all nodes in a distributed systems can configure distributed systems statically

IaaS Provider	Virtual machine types	Private network	Routing configuration	inter-cloud VPN
Amazon EC2	23	yes	yes	yes
Google Compute Engine	15	yes	yes	no
DigitalOcean	9	yes	no	no
RackSpace	12	yes	yes (with special vm)	yes (with special vm)
Azure	10	yes	yes	yes
OpenStack (private)	user defined	yes	yes	yes

Table 1: An overview of features and services in the offering of IaaS providers.

without any manual management overhead and disable discovery mechanisms altogether (failure detection is still required). Therefore, it is trivial to configure a distributed system for single or multi-cloud environment, given that direct communication between all nodes is possible. The configuration management tool generates a static list of IP addresses of other nodes to connect to, and use that list in a configuration file.

Naming. IaaS providers assign DNS names to their virtual machines. However, each provider has their own convention on how this name is generated. Many service rely on a correct and fast DNS infrastructure. In multi-cloud configuration management can configure a set of nameservers (master-slaves) located at each IaaS provider and that can autonomously resolve addresses of all virtual machines. This ensures consistent naming (one authoritative source) and fast lookups (one or more DNS servers at each IaaS). Alternatively, a configuration management tool can also maintain a static hosts file that is deployed on each virtual machine.

Computing. Each IaaS platform provides multiple types of virtual machines, each with different specifications. Easy scaling and migration between different IaaS providers requires a mapping of virtual machine types. A straightforward solution is to define our own virtual machine types in the configuration model and use these types to define virtual machines in the configuration model. Inside the configuration a mapping is provided between our virtual machine types in the configuration model and the types on each IaaS. Based on the selected IaaS provider, the configuration management tool can select the correct virtual machine type to boot a new virtual machine.

4. RELATED WORK

In the current state-of-practice three tools are well known for using a *desired state* configuration model: CFEngine [3], Chef [4] and Puppet [9]. Many companies use these tools to automate configuration management in the cloud. Additionally, software such as Brooklyn [1] or Amazon OpsWorks [11] uses Chef to deploy configuration changes. The problem with these tools is that they compile a configuration model per managed device. This means that it is hard for these tools to generate for example a list of all hostnames and the associated IP-addresses. This information has to be externalised from the configuration model and stored in a database. This results in increased management overhead.

IMP [10] is a research tools from our research group that creates a complete configuration model of the entire infrastructure. It also supports iterative deployment of a configuration model if not all configuration is known beforehand. For example, the IP addresses of virtual machines are only known after they are booted. IMP can first boot virtual machines and when their IP addresses is known, configure the virtual machine and for example, update DNS records. This complete configuration model comes at the price of reduced

scalability because the entire configuration needs to be compiled when only one configuration parameter changes.

5. CONCLUSION

In this paper we argue that adding services and configuration to all virtual machines of an infrastructure, a multi-cloud IaaS can function as a single infrastructure for the upper layers of the execution environment. The key enabling factor for this is extensive automation of deployment and configuration with configuration management tools.

This approach is already possible with existing configuration management tools, albeit with different trade-offs between management overhead and scalability. We illustrated this for a number of challenges associated with multi-cloud. Future work should investigate if other challenges occur when distributed systems are deployed in a multi-cloud IaaS as described in this paper. However, not all challenges can be addressed with configuration. For example, the latency between datacenter or management of application state (such that there is no single point of failure) must be handled higher in the application stack.

6. REFERENCES

- [1] APACHE. Apache brooklyn. <https://brooklyn.incubator.apache.org>, 2014.
- [2] APACHE. Apache jclouds. <https://jclouds.apache.org>, 2014.
- [3] CFENGINE INC. Configuration Management Software. <http://cfengine.com>, 2014.
- [4] CHEF INC. Chef. <http://www.getchef.com>, 2014.
- [5] HAZELCAST. EC2 Auto Discovery. <http://hazelcast.org/docs/latest/manual/html/ec2autodiscovery.html>, 2014.
- [6] HUMMER, W., ROSENBERG, F., OLIVEIRA, F., AND EILAM, T. Testing idempotence for infrastructure as code. In *Middleware '13* (2013).
- [7] JBOSS. Reliable group communication with JGroups. <http://www.jgroups.org/manual/index.html#DiscoveryProtocols>, 2014.
- [8] PATRICK DEBOIS. Infrastructure as code - a comprehensive overview. <http://www.jedi.be/blog/2013/05/24/InfrastructureasCode/>, 2012.
- [9] PUPPET LABS INC. Puppet Labs: IT Automation Software for System Administrators. <http://www.puppetlabs.com>, 2014.
- [10] VANBRABANT, B. *A Framework for Integrated Configuration Management of Distributed Systems*. PhD thesis, June 2014.
- [11] WERNER VOGELS. Expanding the Cloud - Introducing AWS OpsWorks, a Powerful Application Management Solution. <http://www.allthingsdistributed.com/2013/02/aws-opsworks.html>, 02 2013.